



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/482,178	01/12/2000	Theodore M. Osborne, II	1306	6648
22468	7590	01/14/2005	EXAMINER	
CHAPIN & HUANG L.L.C. WESTBOROUGH OFFICE PARK 1700 WEST PARK DRIVE, SUITE 120 WESTBOROUGH, MA 01581			STEELMAN, MARY J	
			ART UNIT	PAPER NUMBER
			2122	

DATE MAILED: 01/14/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/482,178

Applicant(s)

OSBORNE, II ET AL.

Examiner

Mary J. Steelman

Art Unit

2122

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 16 September 2004.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1,3-18 and 20-22 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1,3-18 and 20-22 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This Office Action is in response to Amendment filed 16 September 2004.

Claims 1, 3-18, and 20-22 are pending. Claims 1, 3, 8, 10, 18, 20, and 22 are amended.

Double Patenting

2. Per Applicant's response in Amendment A, filed 01/21/2003, Applicant agrees to file a terminal disclaimer associated with co-pending application, 09/548203, upon indication of allowance.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. **Claims 1, 3-18 and 20-22** are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 5,974,572 to Weinberg et al., in view of U.S. Patent 6,574,578 to Logan.

Weinberg disclosed (Abstract, lines 1-4), "A visual Web site (server) analysis program, implemented as a collection of software components...facilitating the analysis, management (synchronizing...test code) and load-testing of Web sites." Weinberg's invention generates test scenarios (col. 2, lines 38-39 and col. 3, line 7) that emulate multiple concurrent users (running multiple instances of the test code simultaneously) (synchronizing and simultaneously executing) ...on the system. Col. 32, lines 51-68, "As depicted in FIG. 25, multiple Vusers (i.e., multiple

Art Unit: 2122

instances of the Vuser executable) can be run simultaneously on a single workstation...This produces a load in which multiple client requests can...be pending at-a-time (running multiple instances of the test code simultaneously)...”, col. 33, lines 9-14, “To facilitate the formulation of repeatable, multi-Vuser load tests...the user to define a test ‘scenario’ (synchronizing instances of test code) that specifies the details of the test. A scenario may...represent ten users that are concurrently (running multiple instances of the test code simultaneously) attempting to access...”

Weinberg continues to disclose that the Web site components / modules which are most frequently visited are can be written into test scripts (col. 3. 25-35). Col. 5, lines 57-64, “...web site refers generally to...collection of documents and associated data entities...the term may also refer to the associated...software server components used to provide access to such documents...”

Weinberg specified that the content tested is objects (components) (col. 6, lines 8-15) and that a user can retrieve a URL (content object) from the server (col. 10, lines 7-9) (test code accesses ...over said computer network). Weinberg, col. 33, lines 47-50, “...a code module is provided that automatically generates a load testing scenario (test in response to load)...”

Thus Weinberg is load-testing specific modules / components (col. 3, lines 3-4, “...stressed (load-tested) according to typical usage patterns” of an entire Web site application.

Art Unit: 2122

Weinberg did not provide extensive details concerning the testing of individual application technology based object oriented software components of an application under test. However, Logan did disclose more specific details regarding technology based object oriented software components (col. 2, lines 19-22 and col. 6, lines 52-54), "...JAVABeans (technology based object oriented software component)...", and the utilization of a server to coordinate testing. Logan: Abstract, lines 1-3, "Method and system aspects for utilizing a server to coordinate component testing in an integrated test environment network are described."

Per claim 1:

-a) providing test code automatically generated from analysis of the technology based object oriented software component that exercises said technology based object oriented software component of the application under test;

(Logan, col. 6, line 49-col. 7, line 20, "In accordance with a further embodiment of the present invention, an automatic test case code generator is utilized (providing test code automatically generated)... When testing components, e.g., JAVABeans (technology based object oriented software component)...the test case generator generates the skeleton test cases by way of the java.introspection facility (test code automatically generated from analysis of the technology based object oriented software component)...to test the API calls within a class and provides an exhaustive core API unit test case of a given class... Thus, a skeleton test suite program is generated that can then be individualized for specific execution framework to automatically execute each test case (test code provided) and post test results to the common repository.")

Art Unit: 2122

-b) ...wherein said test code accesses said technology based object oriented software component of the application under test over said computer network and recording performance data on said technology based object oriented software component of the application under test;

(Logan, col. 4, lines 45-47, "Once the ITE client is logged on to the ITE server, the desired partition and associated test suites ...are selected...", col. 4, lines 55-57, "The appropriate partitions are then extracted from the partition server and copied to the ITE client (test code accesses said technology based software component of the application under test over said computer network). The ITE client then runs the test suite...", col.6, lines 22-23, "...The test suites are run locally ...and the outcomes of the test cases are stored (recording performance data on said technology based software component of the application under test)...")

-c) repeating step b) multiple times, with a different number of instances of the test code;

(Logan, col.6, line 29, "...repeating an established suite of tests...")

-d) analyzing the recorded performance data to indicate a performance characteristic of said technology based object oriented software component of the application under test in response to load. (Logan, col. 7, line 47-49, "...generates the test results transaction...posted to the database server...")

Logan disclosed (col. 1, lines 24-45) that the 'use of the JAVA programming language' is influencing developers to create components thereby creating 'reusable code'. "...JAVABean components or beans (technology based object oriented software component of the application)

Art Unit: 2122

are reusable software components that can...be combined to create JAVA application or Web-centric applets.” Col. 1, lines 51-53, “Ensuring the integrity of the component product through thorough and high quality testing thus becomes a paramount concern...” Col. 1, line 67-col. 2, line 2, “...a need still exists for an improved approach for more efficient and effective software component testing...” Col. 2, lines 19-22, “The present invention meets these needs and provides method and system aspects for utilizing a server to coordinate component testing in an integrated test environment network.” Thus, Logan is teaching the JAVABean component load-testing, while recognizing that components (col. 1, lines 46-47) may be combined to create JAVA applications or Web-centric applets.” Examiner disagrees, Logan does not teach away from testing applications, as an ‘application’ is generally formed from a combination of components.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Weinberg’s invention which generates a load test on a server by including Logan’s invention which disclosed more details regarding the testing of technology based object oriented software components (col. 2, lines 39-47) in a distributed environment (fig. 1), because extensive testing, the comparison of outcomes, feedback and quality checks of components are useful in software development (Logan, col.7, lines 25-40). All references suggest the testing of components that make up an application.

Per claim 3:

Art Unit: 2122

-application under test is an object oriented language and the step of providing test code comprises providing test code to exercise one object oriented software component of the application. (Weinberg, col. 31, lines 61-67, "An important feature of the invention involves the ability to automatically generate load testing scripts (providing test code)...from server access log files...", col. 33, lines 9-12, "To facilitate the formulation of repeatable...tests...include code for allowing the user to define a test 'scenario' (providing test code) that specifies the details of the test.")

Weinberg disclosed the testing of components, without specifying "an object oriented language" or "object oriented software component" However Logan disclosed more specifically, (col. 1, lines 24-45) the 'use of the JAVA programming language' is influencing developers to create components thereby creating 'reusable code'. "...JAVABean components or beans (technology based object oriented software component of the application) are reusable software components that can...be combined to create JAVA application or Web-centric applets."

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Weinberg's invention which generates a load test on a server by including Logan's invention which disclosed more details regarding the testing of technology based object oriented software components (col. 2, lines 39-47) in a distributed environment (fig. 1), because extensive testing, the comparison of outcomes, feedback and quality checks of components are useful in software development (Logan, col.7, lines 25-40). All references suggest the testing of components that make up an application.

Per claim 4:

-the step of synchronizing comprises starting each instance of the test code at the same time.

(Weinberg, Col. 20, lines 22-26, “A task manager process...handles issues related to the management of the threads, including the synchronization...” The task manager process handles synchronization of the test code.)

Per claim 5:

-the step of synchronizing and executing comprises executing a portion of the plurality of instances of test code on a first computer and a portion of the plurality of instance of test code on a second computer connected to the network. (Weinberg, Fig. 25 and col. 32, lines 51-58, “...multiple Vusers (i.e., multiple instances of the Vuser executable) can be run simultaneously on a single workstation...This produces a load in which multiple client requests can...be pending at-a-time...” See rejection of claim 4 above regarding the synchronization by the task manager process.)

Per claim 6:

Preparing a graphical display having as an independent variable the number of instances of the test code and the dependent variable is the performance data. (Weinberg, Fig. 26 and col. 18, lines 64-67, “...the Action Tracker plug-in communicates with the Web sites...to retrieve server access log files for performing Web site activity analyses. Also, col. 33, lines 3-8, “...user is

Art Unit: 2122

presented with a set of graphical reports (graphical display) that allow the user (to) evaluate the site's performance..."

Per claim 7:

Preparing a graphical display having as an independent variable the number of instances of the test code and the dependent variable is derived from the performance data. (Weinberg, Fig. 26, col. 32, line 64 – col. 33, line 8, "...records various performance-related characteristics of these responses. These characteristics include...response times to individual client requests...the user is presented with a set of graphical reports...the user can...compare response times of different site components..."

Per claim 8:

Weinberg disclosed a Web site (server) analysis program implemented as a collection of software components. Weinberg's invention generates test scenarios (col. 2, lines 38-39 and col. 3, line 7) that emulates multiple concurrent users on the system. Weinberg specified that the content tested is objects (components) (col. 6, lines 8-15) and that a user can retrieve a URL (content object) from the server (col. 10, lines 7-9.)

Weinberg did not provide extensive details concerning the testing of individual application components. However, Logan did disclose:
-application under test is resident on a first server within the network and the application has a remote interface and the test code is resident on at least a second computer within the network

Art Unit: 2122

and exercises the object oriented software component of the application under test using the remote interface of the application under test.

(Logan, col. 7, lines 41-49, "...client instantiates the software objects being tested...passes to ...server..." Also see fig. 1. Additionally Logan provided more details regarding 'object oriented software component of the application under test'. Logan disclosed (col. 1, lines 24-45) that the 'use of the JAVA programming language' is influencing developers to create components thereby creating 'reusable code'. "...JAVABean components or beans (technology based object oriented software component of the application) are reusable software components that can...be combined to create JAVA application or Web-centric applets.")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Weinberg's invention which generates a load test on a server by including Logan's invention which disclosed more details regarding the testing of application components (object oriented software component) (col. 2, lines 39-47) in a distributed environment (fig. 1), because extensive testing of components is useful in comparing expected outcome versus actual outcome, interactive tests can provide feedback, and checklist testing can ensure quality checks are performed in a complete and consistent manner (Logan, col.7, lines 25-40).

Per claim 9:

Art Unit: 2122

-The step of analyzing includes displaying the analyzed data to a human user using a graphical user interface. (Weinberg, Col. 33, lines 3-4, “the user is presented with a set of graphical reports...”)

Per claim 10:

Weinberg disclosed a Web site (server) analysis program implemented as a collection of software components. Weinberg’s invention generates test scenarios (col. 2, lines 38-39 and col. 3, line 7) that emulates multiple concurrent users (simultaneous) on the system. Weinberg specified that the content tested is objects (components) (col. 6, lines 8-15) and that a user can retrieve a URL (content object) from the server (col. 10, lines 7-9.) Thus Weinberg disclosed ‘the load conditions including simultaneously running a plurality of instance of test code automatically generated from analysis of the technology based software component’

Weinberg did not provide extensive details concerning the testing of individual application components, or a user interface. However, Logan did disclose:

-a) specifying test conditions through a user interface to a test system;

(Logan: col. 4, lines 51-52, The ITE client then issues test suite staging requests to the ITE server.”)

Art Unit: 2122

-b) initiating, across a network, through a user interface to the test system the gathering of test data on the performance of at least one technology based object oriented software component of the application under test at a plurality of load conditions...

(Logan, col. 6, line 49-col. 7, line 20, "In accordance with a further embodiment of the present invention, an automatic test case code generator is utilized... When testing components, e.g., JAVABeans (object oriented software components)... the test case generator generates the skeleton test cases by way of the java.introspection facility (test code automatically generated from analysis of the technology based software component)... to test the API calls within a class and provides an exhaustive core API unit test case of a given class... Thus, a skeleton test suite program is generated that can then be individualized for specific execution framework to automatically execute each test case (test code provided) and post test results to the common repository." See Logan, figs. 7 & 11, networked testing. Col. 7, lines 16-20, "Thus, a skeleton test suite program is generated... to automatically execute each test case and post test results (gathering of test data) to the common repository." Col. 7, lines 43-49, "The framework on the ITE client instantiates... objects being tested, drives the execution of test suites, captures responses... generates the test results transaction, which it then passes to the ITE application server to be posted to the database server." (networked transactions) Col. 7, lines 53-55, "AdriverPanel constructs a TestFrame frame window that contains a TestingPanel object (through a user interface) which... contains any graphical elements of the component software being tested...")

-c) specifying through a user interface to the test system the output format of the test data;

Art Unit: 2122

(Logan: col. 2, lines 38-47, "...integrates test environment...Client **user interface**, a data base server, application server, web server and client application code integrates the tools and data include facilities for: developing and executing suites of related test cases, developing and executing testing checklists...and other documentation defining the testing standards (specify output format), methods and procedures and centralized reporting facilities." (emphasis added))

-d) displaying in the specified format the response of said at least one technology based object oriented software component of the application under test to load.

(Logan, col. 7, lines 41-60, and fig. 6, "...AdriverPanel constructs a TestFrame frame window that contains a TestingPanel object which, in turn, contains any graphical elements of the component software being tested..." And Col. 6, lines 52-56, "JAVABeans" (technology based object oriented software component).)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Weinberg's invention which generates a simultaneous load test by including Logan's invention which disclosed more details regarding the testing of application components (technology based object oriented software components) (col. 2, lines 39-47) in a distributed environment (fig. 1), including a user interface, because extensive testing of components, comparing expected outcome versus actual outcome, feedback from interactive tests, and quality checks are desired during software development (Logan, col.7, lines 25-40) and a user interface enhances developer interaction .

Art Unit: 2122

Per claim 11:

-the specified format is a graphical format indicating response time as a function of load conditions. (Weinberg, Col. 33, line 3-4, "...the user is presented with a set of graphical reports...")

Per claim 12:

-the specified graphical format is a Hi-Lo plot. (Weinberg, Col. 33, line 3-4, "...the user is presented with a set of graphical reports...").

Per claim 13:

-the step of gathering data under a plurality of load conditions comprises initiating the execution of a plurality of copies of a test program, with the number of copies executing simultaneously relates to the load condition. (Weinberg, Col. 33, lines 24-39, "To define a scenario, the user initially uses the Web Vuser Generator component to generate the Web scripts to be included within the scenario...the user specifies such details as the number of Vusers, the Web script to be run...and the number of consecutive times...The user can also define one of more Sgroups, and can specify various testing parameters...").

Per claim 14:

-the step of specifying an output format includes specifying a method by which response is measured. (Weinberg, Col. 32. line 64 – col. 33, line 8, "Vuser monitors the Web site's responses to the client requests submitted by that Vuser, and records various performance-related

Art Unit: 2122

characteristics of these responses... These characteristics include, for example, response times... user is presented with a set of graphical reports that allow the user (to) evaluate the site's performance...".)

Per claim 15:

-the step of gathering test data includes recording the execution time between selected points in the test program for each simultaneously executing copy of the test program and analyzing the recorded execution times for all copies of the test program. (Weinberg, Col. 33, line 5-6, "...user can...compare response times of different site components...".)

Per claim 16:

-the step of analyzing comprises determining the average and maximum execution times for each of the load conditions. (Weinberg, Col. 32. line 64 – col. 33, line 8, "Vuser monitors the Web site's responses to the client requests submitted by that Vuser, and records various performance-related characteristics of these responses... These characteristics include, for example, response times... user is presented with a set of graphical reports that allow the user (to) evaluate the site's performance...".)

Per claim 17:

-the computerized application under test comprises software resident on a server controlling access to a computerized database; (Weinberg, Col. 3, lines 5-15, "...based on information stored within a server access log file. The server access log file is ...generated by... Web server.

Art Unit: 2122

These log files contain information about accesses...” Also see FIG. 13, item 180 and col. 25, lines 3-11, “...specifies ...other Web server extension component 180 to which the form is addressed...Astra extract the dataset...”)

-the server is connected to a network and the application under test is simultaneously accessed by a plurality of clients over the network; (Weinberg, FIG. 11 and FIG. 25. See claim 1 for rejection of limitations related to simultaneous testing.)

-the test system is resident on at least a second server connected to the network. (Weinberg, FIGs. 11 & 25)

Per claim 18:

Weinberg disclosed a Web site (server) analysis program implemented as a collection of software components. Weinberg’s invention generates test scenarios (col. 2, lines 38-39 and col. 3, line 7) that emulates multiple concurrent (simultaneous) users on the system. Weinberg specified that the content tested is objects (components) (col. 6, lines 8-15) and that a user can retrieve a URL (content object) from the server (col. 10, lines 7-9). Weinberg disclosed ‘analyzing the recorded times to present information on the performance of the technology based object oriented software component of the application under test as a function of load’ at col. 33, lines 2-4, “Following the load testing process, the user is presented with a set of graphical reports that allow the user evaluate the site’s performance...user can...compare...to identify ...performance problems.”

Weinberg did not provide extensive details concerning the testing of individual application components, technology based object oriented software components, including automatically generated test code. However, Logan did disclose:

-a) providing test code automatically generated from analysis of the technology based object oriented software component to exercise a selected technology based object oriented software component:

(Logan, col. 6, line 49-col. 7, line 20, "In accordance with a further embodiment of the present invention, an automatic test case code generator is utilized...When testing components, e.g., JAVABeans...the test case generator generates the skeleton test cases by way of the java.introspection facility (test code automatically generated from analysis of the technology based object oriented software component)...to test the API calls within a class and provides an exhaustive core API unit test case of a given class...Thus, a skeleton test suite program is generated that can then be individualized for specific execution framework to automatically execute each test case (test code provided) and post test results to the common repository.")

-b) creating a first plurality of copies of the test code; (Logan, col. 6, line 67 – col. 7, line 1, "...test case code generator generates the skeleton test cases by way of the java.introspection facility...")

Art Unit: 2122

-c) ...executing the first plurality of copies of test code while recording times between events in each of the first plurality of copies of test code, wherein said test code accesses said technology based object oriented software component over the computer network; (Logan, col. 7, lines 30-35. See “background automates tests.” FIGs. 1 & 3 show networked accesses.)

-d) creating a second plurality of copies of test code; (Logan, col. 7, lines 16-20, “...a skeleton test suite program is generated that can then be individualized for specific test cases of the component and integrated into the testing execution framework to automatically execute each test case and post test results...”)

-e) ...executing the second plurality of copies of test code while recording times between events in each of the second plurality of copies of test code; (Logan, col. 2, lines 42-47, “Testing tools include facilities for: developing and executing suites of related test cases...”)

-f) repeating a predetermined number of times the steps of creating plural copies of the test code and simultaneously executing the plural copies while recording event times; (Logan, col. 4, lines 57-60, and col. 6, line 29, “...repeating an established suite of tests...”)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Weinberg’s invention which generates a load test on a server by including Logan’s invention which disclosed more details regarding the testing of application components (col. 2, lines 39-47) in a distributed environment (FIG. 1), because extensive testing of

Art Unit: 2122

components is useful in comparing, providing feedback, and ensuring quality (Logan, col.7, lines 25-40).

Per claim 20:

Weinberg disclosed a Web site (server) analysis program implemented as a collection of software components. Weinberg's invention generates test scenarios (col. 2, lines 38-39 and col. 3, line 7) that emulates multiple concurrent users on the system. Weinberg specified that the content tested is objects (components) (col. 6, lines 8-15) and that a user can retrieve a URL (content object) from the server (col. 10, lines 7-9).

Weinberg did not provide extensive details concerning the testing of individual application components. However, Logan did disclose:

- object oriented software component has a plurality of functions therein and the test code exercises functions of the object oriented software components. (Logan, col. 6, lines 61-63, "...generate a basic core API test skeleton program that systematically exposes all protected methods..." Additionally see rejection of claim 1 regarding 'object oriented software components.)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Weinberg's invention which generates a load test on a server by including Logan's invention which disclosed more details regarding the testing of application components (col. 2, lines 39-47) in a distributed environment (FIG. 1), because extensive testing of

Art Unit: 2122

components is useful in comparing expected outcome versus actual outcome, interactive tests can provide feedback, and checklist testing can ensure quality checks are performed in a complete and consistent manner (Logan, col.7, lines 25-40).

Per claim 21:

Weinberg disclosed a Web site (server) analysis program implemented as a collection of software components. Weinberg's invention generates test scenarios (col. 2, lines 38-39 and col. 3, line 7) that emulates multiple concurrent users on the system. Weinberg specified that the content tested is objects (components) (col. 6, lines 8-15) and that a user can retrieve a URL (content object) from the server (col. 10, lines 7-9).

Weinberg did not provide extensive details concerning the testing of individual application components. However, Logan did disclose:

-the events at which times are recorded includes times at which commands are issued to access functions of the software components and times at which execution of the commands are completed. (Logan, col. 7, lines 25-40.)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Weinberg's invention which generates a load test on a server by including Logan's invention which disclosed more details regarding the testing of application components (col. 2, lines 39-47) in a distributed environment (FIG. 1), and recording events because both inventions deal with improving software development, including components in applications

Art Unit: 2122

undergoing tests, and recording some type of results (Weinberg, col. 32, lines 66-67, “records various performance-related characteristics...”).

Per claim 22:

-a) coordination software; (Weinberg, Abstract lines 1-4.)

-b) at least one code generator, receiving as an input commands from the coordination software and having as an output client test code automatically generated from analysis of the technology based object oriented software component;

(Weinberg, Abstract, lines 20-22, “A Load Wizard module uses this activity data to generate testing scenarios...”)

-c) at least one test engine, receiving as an input commands from the coordination software, the test engine comprising a computer server having a plurality of threads thereon, each thread simultaneously executing an instance of the client test code, wherein said client test code accesses said technology based object oriented software component over a computer network;

(Weinberg, FIGs. 25&30. See FIGs. 7 & 11 for examples of network accesses.)

-d) at least one data log having computerized memory, the memory holding timing data created by the instances of the client test code in the plurality of threads; (Weinberg, col. 32, lines 64-67.)

Art Unit: 2122

-e) at least one data analyzer software, operatively connected to the data log, having an output that represents performance of the technology based object oriented software component of the application under test in response to load. (Weinberg, col. 33, lines 3-4, "...user is presented with a set of graphical reports...")

Weinberg disclosed a Web site (server) analysis program implemented as a collection of software components. Weinberg's invention generates test scenarios (col. 2, lines 38-39 and col. 3, line 7) that emulates multiple concurrent users on the system. Weinberg specified that the content tested is objects (components) (col. 6, lines 8-15) and that a user can retrieve a URL (content object) from the server (col. 10, lines 7-9.) Weinberg, Col. 33, lines 47-50, "...a code module is provided that automatically generates a load testing scenario..."

Weinberg did not provide extensive details concerning the testing of individual application components. However, Logan did disclose more specific details regarding application components (technology based object oriented software component) (col. 2, lines 19-22 and col. 6, lines 52-54) and the utilization of a server to coordinate testing. Logan: col. 6, line 49-col. 7, line 20, "In accordance with a further embodiment of the present invention, an automatic test case code generator is utilized...When testing components, e.g., JAVABeans (technology based object oriented software component)...the test case generator generates the skeleton test cases by way of the java.introspection facility (test code automatically generated from analysis of the technology based software component)...to test the API calls within a class and provides an exhaustive core API unit test case of a given class...Thus, a skeleton test suite program is

Art Unit: 2122

generated that can then be individualized for specific execution framework to automatically execute each test case (test code provided) and post test results to the common repository.”

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Weinberg’s invention which generates a load test on a server by including Logan’s invention which disclosed more details regarding the testing of application components (col. 2, lines 39-47) in a distributed environment (FIG. 1), because extensive testing of components is useful in software development. Comparisons, feedback, and quality checks are performed in a complete and consistent manner (Logan, col.7, lines 25-40).

Response to Arguments

(A) Applicants have argued, in substance, the following:

As Applicant has noted on page 9, 2nd paragraph, of Amendment dated 16 September 2004, “Weinberg fails to disclose testing of technology based object oriented software component.”

Examiner’s Response:

Examiner disagrees. The Weinberg reference is not used to provide a suggestion of ‘testing of technology based object oriented software component.’ Weinberg broadly specified that the content tested is objects (components) (col. 6, lines 8-15) and that a user can retrieve a URL (content object) from the server (col. 10, lines 7-9.) More specifically, Logan disclosed (col. 1, lines 24-45) that the ‘use of the JAVA programming language’ is influencing developers to

Art Unit: 2122

create components thereby creating 'reusable code'. "...JAVABean components or beans (technology based object oriented software component of the application) are reusable software components that can...be combined to create JAVA application or Web-centric applets."

(B) Applicants have argued, in substance, the following:

As Applicant has noted on page 9, 2nd paragraph, of Amendment dated 16 September 2004, "Logan fails to disclose testing of a technology based object oriented software component over a network."

Examiner's Response:

Examiner disagrees. Note the language of claim 1 b): "wherein test code **accesses...**component of the application...over said computer network..." (emphasis added) Logan's test code does access the component via a network connection. The ITE (integrated test environment) server provides the code to the ITE client.

Logan, col. 4, lines 45-47, "Once the ITE client is logged on to the ITE server, the desired partition and associated test suites ...are selected...", col. 4, lines 55-57, "The appropriate partitions are then extracted from the partition server and copied to the ITE client (test code accesses said technology based software component of the application under test over said computer network). The ITE client then runs the test suite...", col.6, lines 22-23, "...The test suites are run locally ...and the outcomes of the test cases are stored (recording performance data on said technology based software component of the application under test)..."

Art Unit: 2122

(C) Applicants have argued, in substance, the following:

As Applicant has noted on page 9, 2nd paragraph, of Amendment dated 16 September 2004, “Neither Logan nor Weinberg disclose “load testing a technology based object oriented software component by running multiple instances of the test code simultaneously.”

Examiner’s Response:

Examiner disagrees. Weinberg disclosed (col. 2, line 38) a suggestion to “generate a test that emulates multiple concurrent users...(running multiple instances of the test code simultaneously)”. Col. 32, lines 51-68, “As depicted in FIG. 25, multiple Vusers (i.e., multiple instances of the Vuser executable) can be run simultaneously on a single workstation...This produces a load in which multiple client requests can...be pending at-a-time (running multiple instances of the test code simultaneously)...”, col. 33, lines 9-14, “To facilitate the formulation of repeatable, multi-Vuser load tests...the user to define a test ‘scenario’ that specifies the details of the test. A scenario may...represent ten users that are concurrently (running multiple instances of the test code simultaneously) attempting to access...” Thus, Weinberg does suggest running multiple instances of the test code simultaneously.

(D) Applicants have argued, in substance, the following:

As Applicant has noted on page 9, 1st paragraph, “Logan is teaching away from testing applications while Weinberg discloses testing applications...the references are not properly combinable.”

Art Unit: 2122

Examiner's Response:

In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992).

In this case, Weinberg disclosed (Abstract, lines 1-4), "A visual Web site analysis program, implemented as a collection of software components...facilitating the analysis, management and load-testing of Web sites." Weinberg continues to disclose that the Web site components / modules which are most frequently visited are can be written into test scripts (col. 3. 25-35). Col. 5, lines 57-64, "...web site refers generally to...collection of documents and associated data entities...the term may also refer to the associated...software server components used to provide access to such documents..." Thus Weinberg is load-testing specific modules / components (col. 3, lines 3-4, "...stressed (load-tested) according to typical usage patterns" of an entire Web site application.

Logan disclosed (col. 1, lines 24-45) that the 'use of the JAVA programming language' is influencing developers to create components thereby creating 'reusable code'. "...JAVABean components or beans (technology based object oriented software component of the application) are reusable software components that can...be combined to create JAVA application or Web-

Art Unit: 2122

centric applets.” Col. 1, lines 51-53, “Ensuring the integrity of the component product through thorough and high quality testing thus becomes a paramount concern...” Col. 1, line 67-col. 2, line 2, “...a need still exists for an improved approach for more efficient and effective software component testing...” Col. 2, lines 19-22, “The present invention meets these needs and provides method and system aspects for utilizing a server to coordinate component testing in an integrated test environment network.” Thus, Logan is teaching the JAVABean component load-testing, while recognizing that components (col. 1, lines 46-47) may be combined to create JAVA applications or Web-centric applets.” Examiner disagrees, Logan does not teach away from testing applications, as an ‘application’ is generally formed from a combination of components.

Even the limitations of claim 1 suggest that the ‘technology based object oriented software component’ is a part of an ‘application’ under test.

All references and claim limitations suggest the testing of components that make up an application.

5. Applicant’s amendments and arguments filed 27 May 2004 (paper #21) have been fully considered but they are not persuasive. Examiner maintains the rejection of claims 1, 3-18 and 20-22.

Art Unit: 2122

Conclusion

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Additionally note, IDS filed 31 March 2004:

US Patent 6,473,794 B1 to Guheen et al.

Col. 12, lines 45-65, cols. 60, 66, 67, 127, 130, 131, 175, 271, 272 regarding electronic testing and web site testing tools.

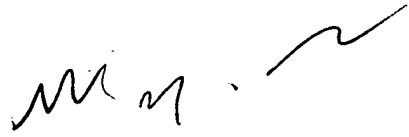
Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (571) 272-3704. The examiner can normally be reached Monday through Thursday, from 7:00 AM to 5:30 PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached at (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Art Unit: 2122

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mary Steelman

01/04/2005



WEI Y. ZHEN
PRIMARY EXAMINER